4. Basics of Database Management System

4.1 Overview of DBMS

- DBMS stands for DataBase Management System and it is a collection of programs that enable users to access databse, manipulate data, report, and represent data.
- It also helps to control access to the database.
- A modern DBMS has the following characteristics
 - 1. Real-world entity A modern DBMS is more realistic and uses real-world entities to design its architecture. It uses behavior and attributes too.
 - 2. Relation-based tables DBMS allows entities and relations among them to form tables.
 - 3. Isolation of data and application A database system is entirely different than its data. A database is an active entity, whereas data is said to be passive, which the database works and organizes.
 - 4. Less redundancy DBMS follows normalization rules, which splits a relation when any of its attributes has redundancy in values.
 - 5. Consistency Consistency is a state where every relation in a database remains consistent. There exist methods and techniques, that can detect attempts of leaving the database in an inconsistent state. A DBMS can provide greater consistency as compared to earlier forms of data-storing applications like file-processing systems.
 - 6. Query Language DBMS is equipped with query language, making retrieving and manipulating data more efficient.
 - 7. ACID Properties DBMS follows the concepts of Atomicity, Consistency, Isolation, and Durability (normally shortened as ACID). These concepts are applied to transactions, which manipulate data in a database. ACID properties help the database stay healthy in multitransactional environments and in case of failure.
 - 8. Multiuser and Concurrent Access DBMS supports multi-user environments and allows them to access and manipulate data in parallel. Though there are restrictions on transactions when users attempt to handle the same data item, users are always unaware of them.
 - 9. Multiple views DBMS offers multiple views for different usersThis feature enables the users to have a concentrated view of the database according to their requirements.
 - 10. Security Features like multiple views offer security to some extent where users are unable to access data of other users and departments. DBMS offers methods to impose constraints while entering data into the database and retrieving the same at a later stage. DBMS offers many different levels of security features, which enables multiple users to have different views with different features Since a DBMS is not saved on the disk as traditional file systems, it is very hard for miscreants to break the code.

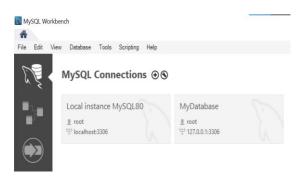
4.2 MySQL

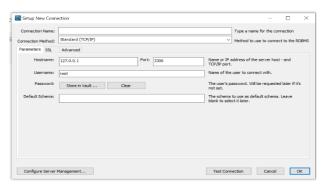
- MySQL is an open-source relational database management system (RDBMS).
- Advantages:
 - 1. MySQL is ideal for both small and large applications
 - 2. MySQL is very fast, reliable, scalable, and easy to use
 - 3. MySQL is cross-platform
 - 4. MySQL is compliant with the ANSI SQL standard
 - 5. MySQL is developed, distributed, and supported by Oracle Corporation.
- Huge websites like Facebook, Twitter, Airbnb, Booking.com, Uber, GitHub, YouTube, etc and Content Management Systems like WordPress, Drupal, Joomla!, Contao, etc. and very large number of web developers around the world uses MySQL.

4.3 Overview of Workbench

- It has basically three main component
 - 1. Connections
 - 2. Models
 - 3. Migration
- Connections:

It shows already exist or used connection with servers or users. We can also add new connection with details like name of connection, port number, authentication details like (SSL key, certificate etc), password etc.

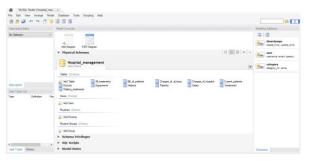


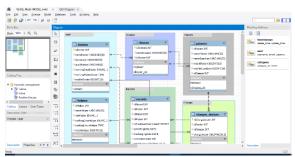


• Model:

It is used to create logical schema of database which gives rich database design and efficient database at end .

It provides various functionality like ERR diagram etc.

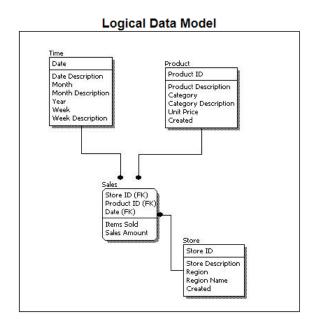


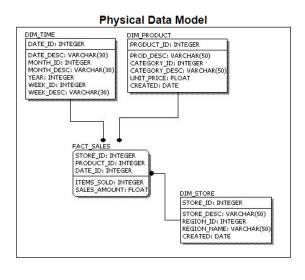


Migration:
 It is used to import other database or to convert other script file like PLSQL or other SQL file into MySQL file.

4.4 Database Design

- Database design can be generally defined as a collection of tasks or processes
 that enhance the designing, development, implementation, and maintenance of
 enterprise data management system. Designing a proper database reduces the
 maintenance cost thereby improving data consistency and the cost-effective
 measures are greatly influenced in terms of disk storage space.
- Designing the database considering the user-based requirements and splitting them out into various models so that load or heavy dependencies on a single aspect are not imposed. Therefore, there has been some model-centric approach and that's where logical and physical models play a crucial role.
 - 1. Physical Model The physical model is concerned with the practices and implementations of the logical model.
 - 2. Logical Model This stage is primarily concerned with developing a model based on the proposed requirements. The entire model is designed on paper without any implementation or adopting DBMS considerations.
- Comparing this physical data model with the logical with the previous logical model, we might conclude the differences that in a physical database entity names are considered table names and attributes are considered column names. Also, the data type of each column is defined in the physical model depending on the actual database used.





4.5 SQL Basics

- SQL stands for Structured Query Language.
- SQL lets you access and manipulate databases.
- SQL statements consists of keywords that are easy to understand.
- The following SQL statement returns all records from a table named "Customers":

SELECT * FROM Customers:

- SQL keywords are NOT case sensitive: 'select' is the same as 'SELECT'
- Semicolon is the standard way to separate each SQL statement in database systems that allow more than one SQL statement to be executed in the same call to the server.
- Some of The Most Important SQL Commands
 - o SELECT extracts data from a database
 - o UPDATE updates data in a database
 - o DELETE deletes data from a database
 - INSERT INTO inserts new data into a database
 - CREATE DATABASE creates a new database
 - o ALTER DATABASE modifies a database
 - o CREATE TABLE creates a new table
 - o ALTER TABLE modifies a table
 - o DROP TABLE deletes a table
 - o CREATE INDEX creates an index (search key)
 - DROP INDEX deletes an index

4.6 Data Sorting

- The ORDER BY keyword is used to sort the result-set in ascending or descending order.
- Syntax

```
SELECT column1, column2, ...
FROM table_name
ORDER BY column1, column2, ... ASC|DESC;
```

- Example
 - Sort the products by price:
 SELECT * FROM Products
 ORDER BY Price;

4.7 Null Value & Keyword

Null values:

- A field with a NULL value is a field with no value.
- If a field in a table is optional, it is possible to insert a new record or update a record without adding a value to this field. Then, the field will be saved with a NULL value.
- Note: A NULL value is different from a zero value or a field that contains spaces. A
 field with a NULL value is one that has been left blank during record creation!
 Keywords:
- This SQL keywords reference contains the reserved words in SQL.

Keyword	Description
ADD	Adds a column in an existing table
ALL	Returns true if all of the subquery values meet the condition
ALTER	Adds, deletes, or modifies columns in a table, or changes the data type of a column in a table
AND	Only includes rows where both conditions is true
ANY	Returns true if any of the subquery values meet the condition
AS	Renames a column or table with an alias
ASC	Sorts the result set in ascending order
BACKUP DATABASE	Creates a back up of an existing database
BETWEEN	Selects values within a given range
CASE	Creates different outputs based on conditions
СНЕСК	A constraint that limits the value that can be placed in a column
COLUMN	Changes the data type of a column or deletes a column in a table
CONSTRAINT	Adds or deletes a constraint
CREATE	Creates a database, index, view, table, or procedure
DEFAULT	A constraint that provides a default value for a column
DELETE	Deletes rows from a table
DESC	Sorts the result set in descending order

DICTING	Calanta and distinct (difference)
DISTINCT	Selects only distinct (different) values
DROP	Deletes a column, constraint, database, index, table, or view
EXISTS	Tests for the existence of any record in a subquery
FOREIGN KEY	A constraint that is a key used to link two tables together
FROM	Specifies which table to select or delete data from
FULL OUTER JOIN	Returns all rows when there is a match in either left table or right table
GROUP BY	Groups the result set (used with aggregate functions: COUNT, MAX, MIN, SUM, AVG)
HAVING	Used instead of WHERE with aggregate functions
IN	Allows you to specify multiple values in a WHERE clause
INDEX	Creates or deletes an index in a table
INNER JOIN	Returns rows that have matching values in both tables
INSERT INTO	Inserts new rows in a table
IS NULL	Tests for empty values
IS NOT NULL	Tests for non-empty values
JOIN	Joins tables
LEFT JOIN	Returns all rows from the left table, and the matching rows from the right table
LIKE	Searches for a specified pattern in a column
LIMIT	Specifies the number of records to return in the result set
NOT	Only includes rows where a condition is not true
NOT NULL	A constraint that enforces a column to not accept NULL values
OR	Includes rows where either condition is true
ORDER BY	Sorts the result set in ascending or descending order
OUTER JOIN	Returns all rows when there is a match in either left table or right table
PRIMARY KEY	A constraint that uniquely identifies each record in a database table
PROCEDURE	A stored procedure

RIGHT JOIN	Returns all rows from the right table, and the matching rows from the left table
ROWNUM	Specifies the number of records to return in the result set
SELECT	Selects data from a database
SELECT DISTINCT	Selects only distinct (different) values
SELECT INTO	Copies data from one table into a new table
SELECT TOP	Specifies the number of records to return in the result set
SET	Specifies which columns and values that should be updated in a table
TABLE	Creates a table, or adds, deletes, or modifies columns in a table, or deletes a table or data inside a table
TOP	Specifies the number of records to return in the result set
TRUNCATE TABLE	Deletes the data inside a table, but not the table itself
UNION	Combines the result set of two or more SELECT statements (only distinct values)
UNION ALL	Combines the result set of two or more SELECT statements (allows duplicate values)
UNIQUE	A constraint that ensures that all values in a column are unique
UPDATE	Updates existing rows in a table
VALUES	Specifies the values of an INSERT INTO statement
VIEW	Creates, updates, or deletes a view
WHERE	Filters a result set to include only records that fulfill a specified condition

4.8 Auto increment

- Auto-increment allows a unique number to be generated automatically when a new record is inserted into a table.
- Often this is the primary key field that we would like to be created automatically every time a new record is inserted.
- Syntax
 - The following SQL statement defines the "Personid" column to be an autoincrement primary key field in the "Persons" table:

```
CREATE TABLE Persons (
Personid int NOT NULL AUTO_INCREMENT,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Age int,
PRIMARY KEY (Personid)
);
```

- By default, the starting value for AUTO_INCREMENT is 1, and it will increment by 1 for each new record.
- To let the AUTO_INCREMENT sequence start with another value, use the following SQL statement:

ALTER TABLE Persons AUTO_INCREMENT=100;

4.9 DDL, DML, DCL, TCL, DQL

DDL (Data Definition Language)

- DDL or Data Definition Language actually consists of the SQL commands that can be used to define the database schema. It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in the database. DDL is a set of SQL commands used to create, modify, and delete database structures but not data. These commands are normally not used by a general user, who should be accessing the database via an application.
- List of DDL commands:
 - CREATE: This command is used to create the database or its objects (like table, index, function, views, store procedure, and triggers).
 - o DROP: This command is used to delete objects from the database.
 - o ALTER: This is used to alter the structure of the database.
 - TRUNCATE: This is used to remove all records from a table, including all spaces allocated for the records are removed.
 - o COMMENT: This is used to add comments to the data dictionary.
 - o RENAME: This is used to rename an object existing in the database.

DQL (Data Query Language)

 DQL statements are used for performing queries on the data within schema objects. The purpose of the DQL Command is to get some schema relation based on the query passed to it. We can define DQL as follows it is a component of SQL statement that allows getting data from the database and imposing order upon it. It includes the SELECT statement. This command allows getting the data out of the database to perform operations with it. When a SELECT is fired against a table or tables the result is compiled into a further temporary table, which is displayed or perhaps received by the program i.e. a front-end.

- List of DQL:
 - SELECT: It is used to retrieve data from the database.

DML(Data Manipulation Language)

- The SQL commands that deal with the manipulation of data present in the
 database belong to DML or Data Manipulation Language and this includes most
 of the SQL statements. It is the component of the SQL statement that controls
 access to data and to the database. Basically, DCL statements are grouped with
 DML statements.
- List of DML commands:
 - o INSERT: It is used to insert data into a table.
 - o UPDATE: It is used to update existing data within a table.
 - o DELETE: It is used to delete records from a database table.
 - o LOCK: Table control concurrency.

DCL (Data Control Language)

- DCL includes commands such as GRANT and REVOKE which mainly deal with the rights, permissions, and other controls of the database system.
- List of DCL commands:
- GRANT: This command gives users access privileges to the database.
- Syntax:
 - GRANT SELECT, UPDATE ON MY_TABLE TO SOME_USER, ANOTHER_USER
 :
- REVOKE: This command withdraws the user's access privileges given by using the GRANT command.
- Syntax:
 - REVOKE SELECT, UPDATE ON MY_TABLE FROM USER1, USER2;

TCL (Transaction Control Language)

- Transactions group a set of tasks into a single execution unit. Each transaction
 begins with a specific task and ends when all the tasks in the group are
 successfully completed. If any of the tasks fail, the transaction fails. Therefore, a
 transaction has only two results: success or failure. You can explore more about
 transactions here. Hence, the following TCL commands are used to control the
 execution of a transaction:
 - o BEGIN: Opens a Transaction.
 - o COMMIT: Commits a Transaction.
- Syntax:
 - o COMMIT;
- ROLLBACK: Rollbacks a transaction in case of any error occurs.
- Syntax:
 - o ROLLBACK;
- SAVEPOINT: Sets a save point within a transaction.
- Syntax:
 - SAVEPOINT SAVEPOINT_NAME;

4.10 Limit

- The LIMIT clause is used to specify the number of records to return.
- The LIMIT clause is useful on large tables with thousands of records. Returning a large number of records can impact performance.
- LIMIT Syntax
 - SELECT column_name(s)

FROM table_name

WHERE condition

LIMIT number:

- If we want to select records inclusive MySQL provides a way to handle this: by using OFFSET.
- Example
 - SELECT * FROM Customers

LIMIT 3 OFFSET 3;

SELECT * FROM Customers

LIMIT 3,4; (3 is offset and 4 is no. of records to be selected)

4.11 Aggregate functions

- Aggregate function is used to perform calculations on multiple values and return
 the result in a single value like the average of all values, the sum of all values, and
 maximum & minimum value among certain groups of values. We mostly use the
 aggregate functions with SELECT statements in the data query languages.
- Syntax:
 - function_name (DISTINCT | ALL expression)
- First, we need to specify the name of the aggregate function.
- Second, we use the DISTINCT modifier when we want to calculate the result based on distinct values or ALL modifiers when we calculate all values, including duplicates. The default is ALL.
- Third, we need to specify the expression that involves columns and arithmetic operators.
- Some of the most commonly used aggregate functions are in the below table:

Aggregate Function	Descriptions
count()	It returns the number of rows, including rows with NULL values in a
	group.
sum()	It returns the total summed values (Non-NULL) in a set.
average()	It returns the average value of an expression.
min()	It returns the minimum (lowest) value in a set.
max()	It returns the maximum (highest) value in a set.

4.12 Sub- Queries

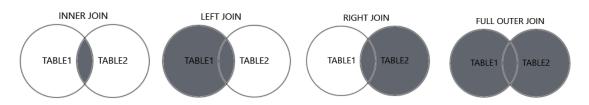
- A subquery in MySQL is a query, which is nested into another SQL query and embedded with SELECT, INSERT, UPDATE or DELETE statement along with the various operators. We can also nest the subquery with another subquery.
- A subquery is known as the inner query, and the query that contains subquery is known as the outer query.
- The inner query executed first gives the result to the outer query, and then the main/outer query will be performed. MySQL allows us to use subquery anywhere, but it must be closed within parenthesis.
- Subqueries should always use in parentheses.
- If the main query does not have multiple columns for subquery, then a subquery can have only one column in the SELECT command.

- We can use various comparison operators with the subquery, such as >, <, =, IN,
 ANY, SOME, and ALL. A multiple-row operator is very useful when the subquery
 returns more than one row.
- We cannot use the ORDER BY clause in a subquery, although it can be used inside the main query.
- If we use a subquery in a set function, it cannot be immediately enclosed in a set function.
- The following are the advantages of using subqueries:
 - The subqueries make the queries in a structured form that allows us to isolate each part of a statement.
 - The subqueries provide alternative ways to query the data from the table;
 otherwise, we need to use complex joins and unions.
 - The subqueries are more readable than complex join or union statements.
- MySQL Subquery Syntax

SELECT column_list (s) FROM table_name
WHERE column_name OPERATOR
(SELECT column_list (s) FROM table_name [WHERE])

4.13 JOINS

- The SQL Join clause is used to combine data from two or more tables in a database.
 When the related data is stored across multiple tables, joins help you to retrieve records combining the fields from these tables using their foreign keys.
- Here are the different types of the JOINs in SQL:
 - o INNER JOIN: Returns records that have matching values in both tables
 - LEFT (OUTER) JOIN: Returns all records from the left table, and the matched records from the right table
 - RIGHT (OUTER) JOIN: Returns all records from the right table, and the matched records from the left table
 - FULL (OUTER) JOIN: Returns all records when there is a match in either left or right table



4.14 UNIONS

- MySQL Union is an operator that allows us to combine two or more results from
 multiple SELECT queries into a single result set. It comes with a default feature
 that removes the duplicate rows from the result set. MySQL always uses the
 name of the column in the first SELECT statement will be the column names of
 the result set(output).
- MySQL Union must follow these basic rules:
 - The number and order of the columns should be the same in all tables that you are going to use.
 - The data type must be compatible with the corresponding positions of each select query.
- Syntax

SELECT column_list FROM table1

UNION

SELECT column_list FROM table2;

• We can understand the Union operator with the following visual representation:

Table 1	
Column 1	Column 2
Α	1
Α	2
А	3

Table 2	
Column 1	Column 2
Α	1
В	2
С	3

Table 1 Union Table 2			
Column 1	Column 2		
Α	1		
Α	2		
Α	3		
В	2		
С	3		

4.15 INDEX

- A database index is a data structure that improves the speed of operations in a table. Indexes can be created using one or more columns, providing the basis for both rapid random lookups and efficient ordering of access to records.
- While creating index, it should be taken into consideration which all columns will be used to make SQL queries and create one or more indexes on those columns.
- Practically, indexes are also a type of tables, which keep primary key or index field and a pointer to each record into the actual table.
- The users cannot see the indexes, they are just used to speed up queries and will be used by the Database Search Engine to locate records very fast.
- The INSERT and UPDATE statements take more time on tables having indexes,
 whereas the SELECT statements become fast on those tables. The reason is that

while doing insert or update, a database needs to insert or update the index values as well.

- We can create a unique index on a table. A unique index means that two rows cannot have the same index value.
- Syntax:

CREATE UNIQUE INDEX index_name ON table_name (column1, column2,...);

4.16 View

- In SQL, a view is a virtual table based on the result-set of an SQL statement.
- A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.
- You can add SQL statements and functions to a view and present the data as if the data were coming from one single table.
- CREATE VIEW
 - o A view is created with the CREATE VIEW statement.
 - Syntax

CREATE VIEW view_name AS

SELECT column1, column2, ...

FROM table name

WHERE condition;

or

CREATE [OR REPLACE] VIEW view_name AS

SELECT columns

FROM tables

[WHERE conditions];

OR REPLACE: It is optional. It is used when a VIEW already exists. If you
do not specify this clause and the VIEW already exists, the CREATE VIEW
statement will return an error.

UPDATE VIEW

- In MYSQL, the ALTER VIEW statement is used to modify or update the already created VIEW without dropping it.
- Syntax:

ALTER VIEW view_name AS

SELECT columns

FROM table

WHERE conditions:

DROP VIEW

- We can drop the existing VIEW by using the DROP VIEW statement.
- Syntax:DROP VIEW [IF EXISTS] view name;
- IF EXISTS: It is optional. If we do not specify this clause and the VIEW doesn't exist, the DROP VIEW statement will return an error.

4.17 Backup & Restore, Explain Keyword

- The SQL Server backup and restore technique involves steps for creating backups, restoring databases, and saving a valid backup copy in a different location. It is a built in SQL Server feature, useful for disaster recovery plans.
- The backup and restore process in SQL Server involves three main steps:
 - 1. Backing up the desired SQL Server database
 - 2. Transferring the backup files to another location, usually a remote location to avoid risk of a hard drive failure along with possible software problems (e.g. virus attacks)
 - 3. Restoring the database on SQL Server to ensure that the database backup is valid and to verify integrity.

