DQL(Data Query Language):

1) **What is SQL(Structured Query Language)?**
2) Capabilities of **SELECT** statement:
    1) Projection:
        1) Used to choose the columns in a table that you want returned by your query.
        2) Can be used to choose a few or many columns of the table as you require
    2) Selection:
        1) Used to choose the rows in a table that you want to be returned by the query
        2) We can restrict the rows by using WHERE clause
    3) Joining
3) Steps while writing SQL statements:
    1) NOT case-sensitive
    2) Can be of one or more lines
    3) Keywords cannot be abbreviated or split across lines
    4) Clauses are placed in separate lines
    5) Indents are used to enhance the readability
4) Aliasing Ways:
    1) Using AS keyword:
        1) SELECT dept_id AS “Dept No” ———————;
    2) Without using AS and using Double Quotes:
        1) SELECT dept_id “DEPT NO” ———————;
    3) Without using AS and without using Double Quotes:
        1) In this case we cannot alias the column name with spaces we need to add underscores in place of spaces
        2) EG: SELECT dept_id DEPT_NO ———————-;
    4) Using AS and without using Double Quotes:
        1) EG: SELECT dept_id AS DEPT_NO ———————-;
5) Arithmetic Operators:
    1) We can use them in SELECT and WHERE clause but n**ot in FROM clause**
    2) Precedence Order:
        1) *, /, +, -
6) Operators used inside WHERE clause:
    1) To combine multiple condition:
        1) OR
        2) AND
    2) Comparison Operators:
        1) >, <, >=, <=, <>{not equals to}
    3) Range-Search Conditions:
        1) BETWEEN, NOT BETWEEN

2) BETWEEN <<lower-range>> AND <<upper-range>>;
    1) [lower-range, upper-range] -> both the ranges are inclusive
4) Set-Membership search conditions:
    1) Used to fetch values in specified set
    2) IN, NOT IN
5) Pattern-Match Search Condition:
    1) use LIKE keyword
    2) Wild-card search:
        1) %: represents any sequence of 0 or more characters
            1) EG:
                1) Starts with a -> “a%”
                2) Ends with a -> “%a”
                3) Contains a -> “%a%”
        2) _: represents any single character
6) NULL search conditions:
    1) IS NULL
    2) IS NOT NULL
7) ORDER BY:
    1) Sorts the rows
    2) Modes:
        1) ASC: ascending-> By Default
        2) DESC: descending
    3) The last clause in the select keyword
    4) Eg:
        1) select * from student where name LIKE "J%" order by age;
        2) select * from student where name LIKE "J%" order by age DESC;
        3) select * from employees order by dept_id DESC, salary;
            1) In the above example first my data will be sorted in descending order based upon dept_id and then inside dept_id the data will be sorted based on salary in ascending order
7) MYSQL Functions:
    1) Group Functions:
        1) They operate on a set of rows to give one result per group
        2) Group Functions:
            1) Eg:
                1) Select sum(salary) from employees;
                2) Select avg(salary) from employees;
                3) Select count(*) from employees; -> It will count the rows even if some value is null or duplicate
                4) Select count(dept_id) from employees; -> it will only give count of non-null values
                5) Select max(salary) from employees;
                6) Select min(salary) from employees;

7) Select count(distinct(dept_id)) from employees;

3) Creating Groups:

1) Group By - Having Clause:

1) Eg:
   1) select dept_id, avg(salary) from employees group by dept_id having dept_id IS NOT NULL;
2) Guidelines for GROUP-BY clause:
   1) All columns in the select list that are not in the group function must be inside the group-by clause.
   2) EG: Below will throw error
      1) select last_name, dept_id, avg(salary) from employees group by dept_id having dept_id IS NOT NULL;
3) We cannot use column aliases inside group-by clause:
   1) EG:
      1) select last_name, dept_id, avg(salary) from employees group by dept_id AS "DEPT", last_name AS "LAST" having dept_id IS NOT NULL AND last_name IS NOT NULL;
4) Group By clause can only be used with group/aggregate functions like: SUM, AVG, MIN, MAX, COUNT etc.
5) Aggregate/Group Functions cannot be used inside GROUP-BY clause, but we can use them inside HAVING clause
   1) select dept_id, SUM(salary) from employees GROUP BY dept_id HAVING sum(salary)>3700 AND dept_id IS NOT NULL;

2) COMPLETE SYNTAX OF SELECT CLAUSE:

1) SELECT *, <<group-functions>>
   FROM <<table-name>>
   WHERE <<conditions>>
   GROUP BY <<col-name>>
   HAVING <<conditions>>
   ORDER BY <<ASC|DESC>>
2) What Happens:
   1) The rows are filtered based on the WHERE clause
   2) Groups are made from the filtered rows using the GROUP BY clause
   3) Having clause filters the created groups
3) EG:
   1) SELECT dept_id, sum(salary)
      FROM employees
      WHERE last_name LIKE '_a%'
      GROUP BY dept_id
      HAVING sum(salary)>2000 AND dept_id IS NOT NULL
      ORDER BY dept_id;