Ex. No.: 10

Date:

**Database with VIEW and INDEX**

**AIM:**

To create view and index on the given relation.

**DESCRIPTION:**

**VIEW:**

MySQL has supported database views since version 5+. In MySQL, almost features of views conform to the SQL: 2003 standard. MySQL processes query against the views in two ways:

1. In a first way, MySQL creates a temporary table based on the view definition

2. Statement and executes the incoming query on this temporary table.

3. In a second way, MySQL combines the incoming query with the query defined the view into one query and executes the combined query.

**SYNTAX- VIEW:**

**CREATE** [ALGORITHM = {MERGE | TEMPTABLE | UNDEFINED}]

**VIEW** [**database\_name**].[view\_name]

**AS** [SELECT statement]

**INDEX:**

A database index, or just index, helps **speed up the retrieval of data from tables**. When you query data from a table, first MySQL **checks if the indexes exist**, then MySQL uses the indexes to select exact physical corresponding rows of the table instead of scanning the whole table..

**SYNTAX- INDEX:**

**CREATE** [UNIQUE|FULLTEXT|SPATIAL] **INDEX** index\_name

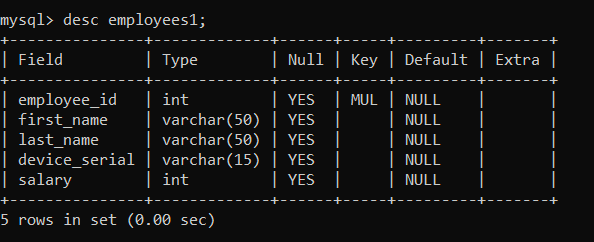
**USING** [BTREE | HASH | RTREE]

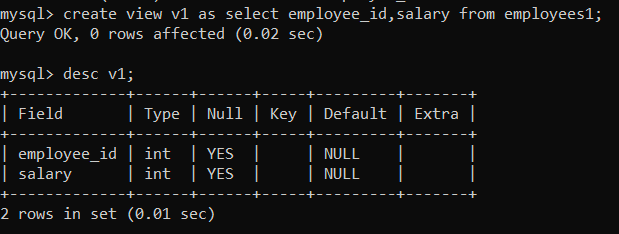
**ON** table\_name (column\_name [(length)] [ASC | DESC],...)

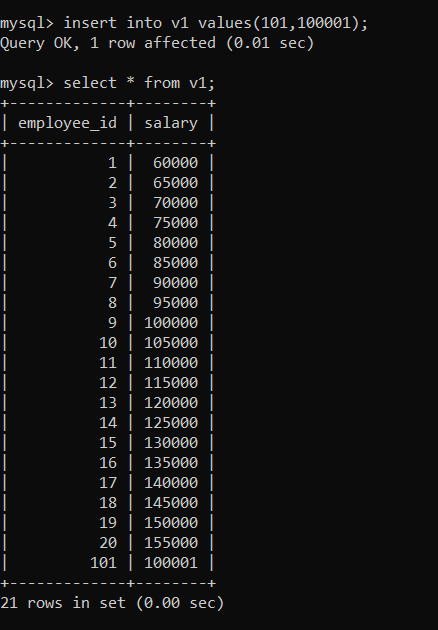
**Questions on View:**

1. Create a view with name ‘v1’ using employees1 table which holds the value of employee\_id and salary of employee.
2. Do the insert and delete records from v1 table.

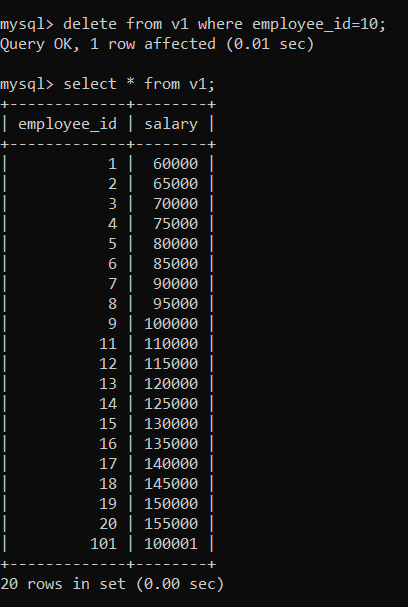


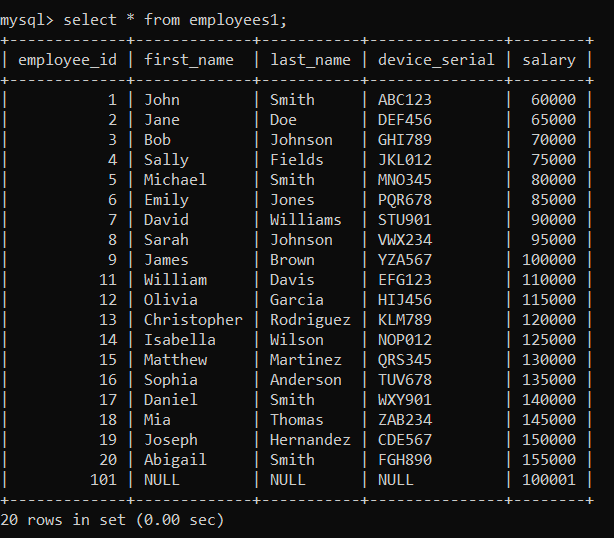








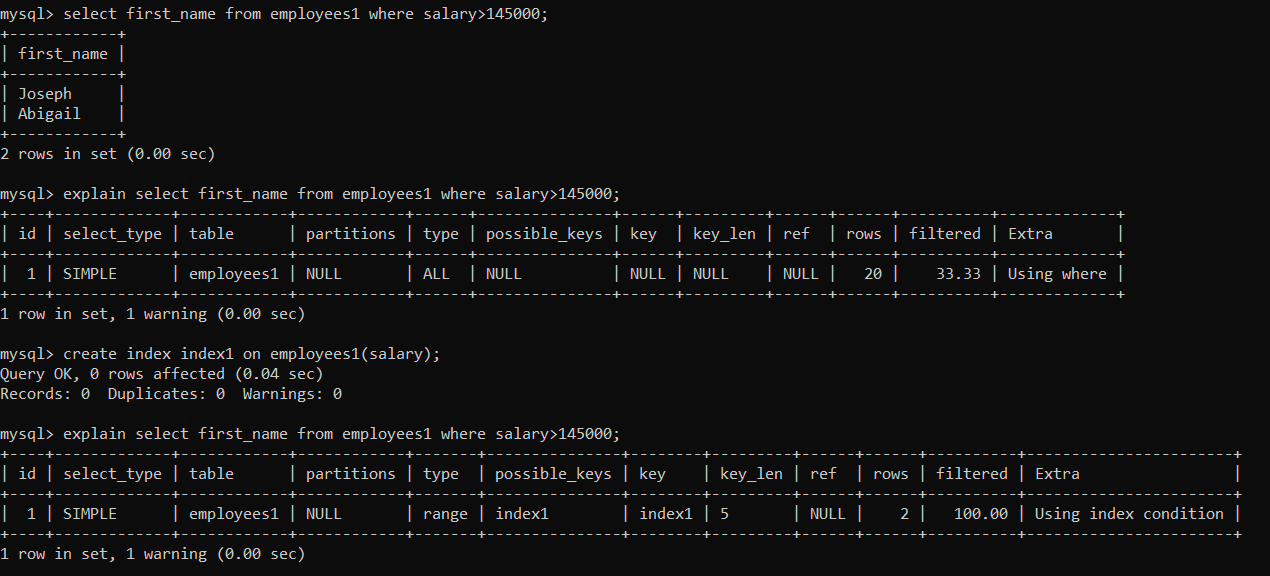


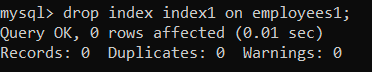


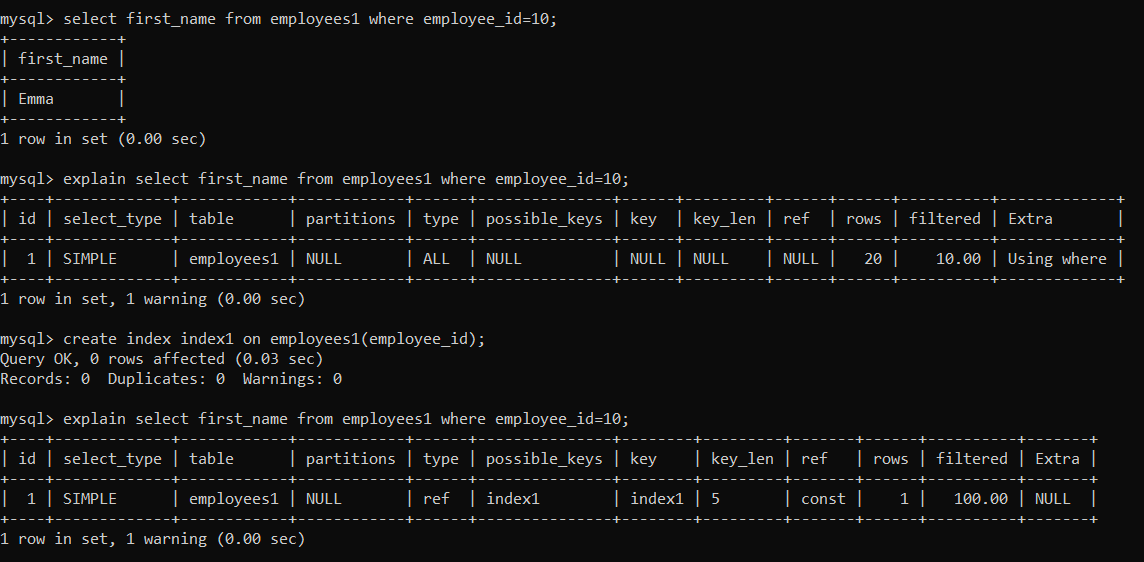
**Questions on INDEX:**

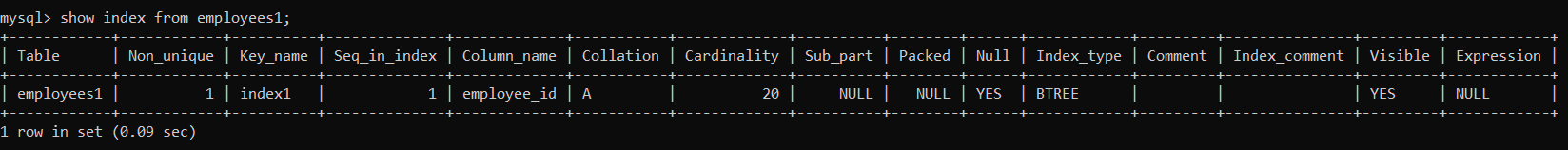
1. Create index1 for ‘salary’ attribute from employees1 relation and list the first name of the employees whose salary is above 145000 and explain the working principle of indexing and then drop the index1.
2. Create index1 for ‘employee\_id’ attribute and display the first name of an employee whose employee id is 10 and explain the working principle of index1.











**RESULT**:

The records from the tables are displayed using view and index on the given relation.