

R.M.K

COLLEGE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Date: 28-04-20
to 12-04-20

Target audience : II
Year CSE Students

Facilitator:
P.N.Senthil Prakash, AP/CSE

Convenor: Dr.D.Paulraj,
Prof. Head of CSE Department

Topic : Training in SQL **(Hands on session)**

Summary:

The two weeks training program aimed at providing hands on experience in SQL. Various topics including simple queries, operators like 'IN', 'NOT IN', 'LIKE', 'HAVING' etc, aggregation functions, views, nested queries, joins, PL/SQL procedures & triggers were discussed during this training session. Students were trained using mysql database and practice exercises were given at end of each session.

Tools Used

MYSQL:

MYSQL is a database management system that allows us to manage relational databases. It is open source software backed by Oracle. It means we can use MySQL without paying a dime. Also, if we want, we can change its source code to suit our needs. Even though MySQL is open source software, we can buy a commercial license version from Oracle to get premium support services. MySQL is pretty easy to master in comparison with other database software like Oracle Database, or Microsoft SQL Server. MySQL can run on various platforms UNIX, Linux, Windows, etc. We can install it on a server or even in a desktop.

Topics Discussed:

Views:

Views in SQL are considered as a virtual table. A view also contains rows and columns. To create the view, we can select the fields from one or more tables present in the database. A view can either have specific rows based on certain condition or all the rows of a table.

Joins:

The SQL Joins clause is used to combine records from two or more tables in a database. A JOIN is a means for combining fields from two tables by using values common to each. The different types of joins are inner join, left outer join, right outer join and full outer joins.

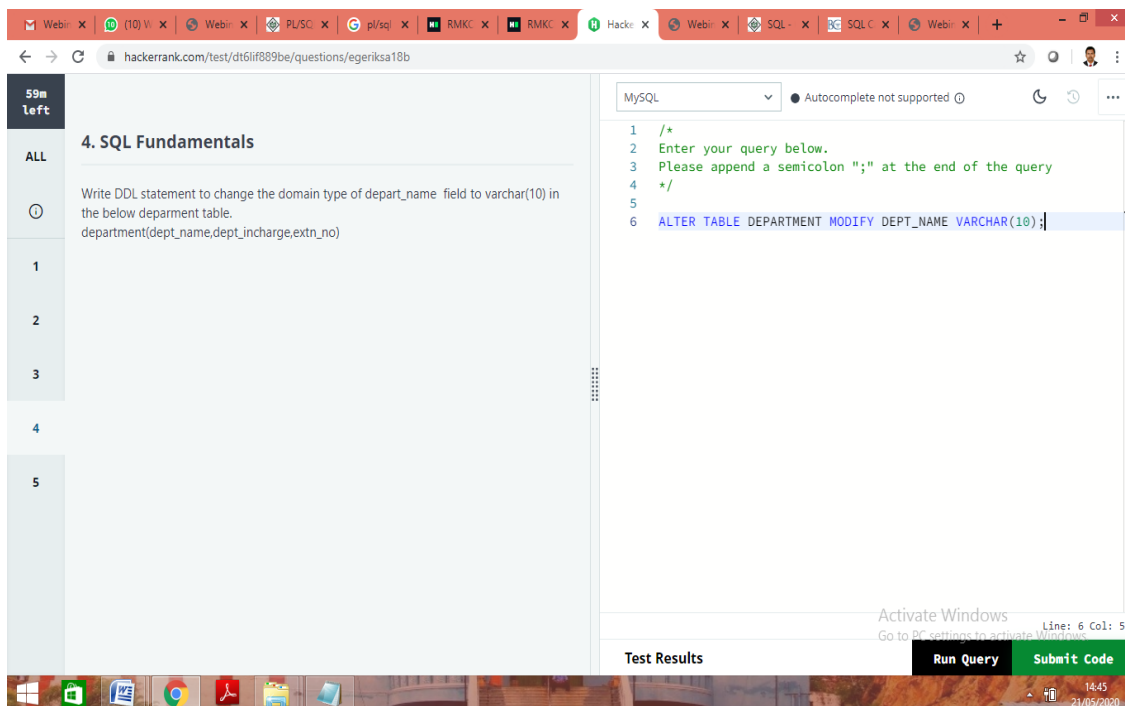
Cartesian Product:

The CARTESIAN JOIN or CROSS JOIN returns the Cartesian product of the sets of records from two or more joined tables. Thus, it equates to an inner join where the join-condition always evaluates to either True or where the join-condition is absent from the statement.

Summary of the technical session.

The participants were trained in the following features in SQL.

- Basic SQL commands – DDL, DML, TCL & DCL.



The screenshot shows a web browser window with the URL hackerrank.com/test/dt6if89be/questions/egeriksa18b. The page title is "4. SQL Fundamentals". The question text is: "Write DDL statement to change the domain type of depart_name field to varchar(10) in the below department table. department(dept_name,dept_incharge,extr_no)". The SQL editor shows the following code:

```
1 /*
2 Enter your query below.
3 Please append a semicolon ";" at the end of the query
4 */
5
6 ALTER TABLE DEPARTMENT MODIFY DEPT_NAME VARCHAR(10);
```

At the bottom of the editor, there are buttons for "Run Query" and "Submit Code". The status bar at the bottom right shows "Line: 6 Col: 53" and "21/05/2020 14:45".

- Simple SQL queries, queries using aggregation functions.
- Creating & manipulating views.
- Queries using Cartesian product and joins.

The screenshot shows a HackerRank interface for a SQL question titled "4. SQL Joins". The question asks to list the names and email IDs of employees in the "Chennai" location, ordered by hire date. The provided schema includes tables for departments, employees, and locations. The SQL editor contains the following query:

```
1 /*
2 Enter your query below.
3 Please append a semicolon ";" at the end of the query
4 */
5
6 SELECT EMP_NAME,EMAIL FROM EMPLOYEES E, DEPARTMENT D,
LOCATION L WHERE E.DEPT_ID=D.DEPT_ID AND D.LOC_ID=L.LOC_IDAND
CITY='CHENNAI' ORDER BY HIRE_DATE;
```

- To create procedures and triggers for real-time applications.

The screenshot shows a HackerRank interface for a PL/SQL question titled "3. PL/SQL Triggers". The question asks to create a trigger that inserts records into a backup table whenever a record is inserted into the main table. The provided schema includes tables for transaction and transaction backup. The SQL editor contains the following query:

```
1 > delimiter $$...
2
3
4 CREATE TRIGGER TRANS_INSRT AFTER INSERT ON TRANSACTION
5 FOR EACH ROWS
6
7 BEGIN
8
9 INSERT INTO TRANSACTION_BACKUP VALUES(NEW.ACCNO,
NEW.TRANS_TYPE,NEW.AMOUNT,NEW.DATE);
10
11 END
12 $$
13 > ...
```