MES College of Engineering Pune-01

Department of Computer Engineering

Name of Student:	Sudesh Pawar	Class:	TE Comp I
Semester/Year:	V / 3rd	Roll No:	25
Date of Performance:	25/07/2019	Date of Submission:	29/07/2019
Examined By:		Experiment No: Part A-04	

GROUP: B ASSIGNMENT NO: 04

AIM: Design at least 10 SQL queries for suitable database application using SQL DML statements:all types of Join, Sub-Query and View..

OBJECTIVES:

- To develop basic, intermediate and advanced Database programming skills.
- To develop basic Database administration skill.

APPRATUS:

- Operating System recommended: 64-bit Open source Linux or its derivative
- Front End: Java/PHP/Python
- Backend: MySql 5.5

IMPLEMENTATION:

1. Create following Tables

cust_mstr(cust_no,fname,lname)

add_dets(code_no,add1,add2,state,city,pincode)

Retrieve the address of customer Fname as 'xyz' and Lname as 'pqr'

2.Create following Tables

cust_mstr(custno,fname,lname)

acc_fd_cust_dets(codeno,acc_fd_no)

fd_dets(fd_sr_no,amt)

List the customer holding fixed deposit of amount more than 5000

Group A: Assignment No: 4

Aim: Design at least 10 SQL queries for suitable database application using SQL DML statements all types of Join, Sub-Query and View.

Questions:

1. What is Join Clause?

→ A JOIN clause is used to combine rows from two or more tables, based on a related column between the tables.

→ A related column between the tables is important for the JOIN clause.

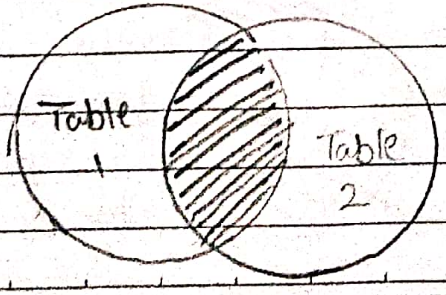
2. What are the different kinds of Joins? Explain in detail.

Ans. i) INNER JOIN:

The INNER JOIN keyword selects records that have matching values in both tables.

Syntax:

```
SELECT column_name(s) FROM table1
INNER JOIN table2 ON table1.column_name =
                    table2.column_name
```



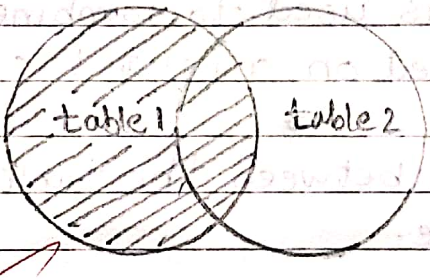
ii) LEFT OUTER JOIN

→ The LEFT OUTER JOIN keyword returns all records from the left table & the matched records from the right table.

→ The result is NULL from the right side, if there is no match.

Syntax:

```
 SELECT column_name(s) FROM table1
LEFT OUTER JOIN table2 ON table1.column_name =
                          table2.column_name.
```

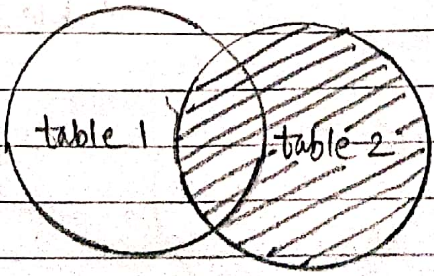


iii) RIGHT OUTER JOIN

The RIGHT OUTER JOIN keyword returns all records from the right table, and the matched records from the left side. (left table). The result is NULL from left side, when there is no match

Syntax:

```
SELECT column_name(s) FROM table1 RIGHT OUTER JOIN
table2 ON table1.column_name=table2.column_name.
```

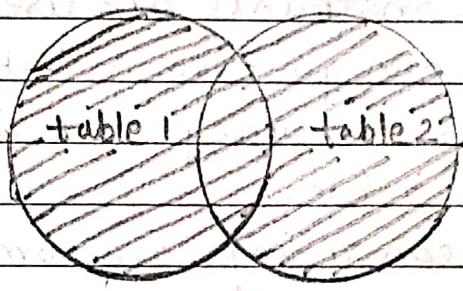


iv) FULL OUTER JOIN

The FULL OUTER JOIN keyword returns all records when there is a match in either left or right table records

Syntax:

```
    SELECT column_name(s) FROM table1
FULL OUTER JOIN table2  ON
    table1.column_name = table2.column_name
```



3) What is cross join?

Ans. The CROSS JOIN produces a result set which is the product of rows of 2 associated tables when no WHERE clause is used with CROSS JOIN.

-> This kind of result is also known as Cartesian Product.

Syntax:

```
    SELECT * FROM table1 CROSS JOIN table2 ;
```

4) What is NULL Value? How is it different from zero value?

Ans. The NULL value is used to represent a missing value. A NULL value, in a table is the value in a field that appears to be blank. A field with zero value is not blank and therefore, a NULL value is different from a zero value or a field that contains spaces. It is not possible to test for NULL values with comparison operators.

5. What are different MySQL constraints?

Ans. Constraints are used to specify rules for data in a table. Constraints are used to limit the type of data that can go into a table.

→ This ensures accuracy & reliability of the data in the table.

The following constraints are commonly used in MySQL:

i) NOT NULL : Ensures that column cannot have a NULL Value.

ii) UNIQUE : Ensures that all values in a column are different.

iii) PRIMARY KEY: A combination of NOT NULL and UNIQUE which uniquely identifies each row in table.

iv) FOREIGN KEY: Uniquely identifies a row/record in another table.

v) DEFAULT: Sets a default value for a column when no value is specified.

6. What is difference between Join and Union ?

Ans.

JOIN	UNION
i) JOIN combines attributes of the tuples present in 2 different relations that share some common attributes.	i) UNION combines tuples of the relations that are present in the query.
ii) JOIN is applicable when the 2 involved relations have atleast one common attribute.	ii) UNION is applicable when the no. of columns present in query are same & the corresponding attributes has same domain.
iii) Types: INNER, FULL(OUTER), LEFT, RIGHT	iii) TYPES: UNION, UNION ALL.
iv) The length of the resultant tuples is more as compared to length of tuples of involved relations.	iv) The no. of resultant tuples is more as compared to no. of tuples present in each relation involved in query.

7) List different aggregation functions in MySQL.

Ans. Different aggregation functions in MySQL are:

i) AVG() - returns the average value of the argument.

ii) COUNT() - returns count of no. of rows.

iii) MAX() - returns max value

iv) MIN() returns minimum value

v) SUM() - returns the sum.

8) What is difference between Unique key & Primary Key?

Ans.

	Primary Key.		Unique Key.
i)	Primary Key can't accept NULL Values.	i)	Unique can accept only one NULL value.
ii)	By default, primary key is clustered index.	ii)	By default, unique key is non-clustered index.
iii)	We can have only one primary key in a table	iii)	We can have more than one unique key in table.
iv)	Primary Key can be made foreign key into another table.	iv)	Unique key can be made foreign key into another table.

9) Explain WHERE and HAVING clause

Ans. WHERE -

WHERE clause is used to filter only those records that are fulfilled by a specific condition given by the user.

In other words, WHERE clause is used to restrict the no. of rows affected by a SELECT, UPDATE or DELETE query.

Syntax:

```
SELECT column1, column2 FROM table-name
WHERE condition.
```

HAVING:

The HAVING clause is used in the SELECT statement to specify filter conditions for a group of rows or aggregates. The HAVING clause is often used with the GROUP BY clause to filter groups based on a specific condition. If the GROUP BY clause is omitted, the HAVING clause behaves like WHERE clause.

Syntax:

```
SELECT column_name(s) FROM table_name
WHERE condition. GROUP BY column_name(s)
HAVING condition ORDER BY column_name(s);
```

C