

Pune Institute of Computer Technology, Pune

Department of Computer Engineering

A.Y. 2020-21 Semester: I

Database Management System Lab

Name: Aditya Sawant

Roll No: 31302

Batch: K3

#### ASSIGNMENT 4

TITLE:

Design at least 10 SQL Queries for suitable database application using MYSQL (All types of Join , Sub-Query and View)

PROBLEM DEFINITION:

Execute the following queries:

1. Find the professor details and department details using NATURAL JOIN.
2. Find the prof\_id, prof\_name and shift.
3. List all the department details and the corresponding names of professors in the same department.
4. List all the professors and the corresponding names of department.
5. Display professor name, dept\_name, shift, salary where prof\_id = 101;
6. list the total number of professor in each department.
7. List the prof\_id associated department and the dept\_name having name 'computer';
8. Find the names of all departments where the professors joined in year 2015 (or date of joining is 1-1-2015).
9. Create view showing the professor and shift details.
10. Perform Manipulation on simple view-Insert, update, delete, drop view.

OBJECTIVE:

To understand

- Types of joins,
- Subquery and its types,

- Complex views

#### OUTCOME:

We will be able to understand types of joins , subquery and its types and complex views and implement MYSQL Queries for suitable database application

#### HARDWARE REQUIREMENTS:

- MONITOR
- KEYBOARD
- 2GB RAM
- 2.4GHz I5 PROCESSOR

#### SOFTWARE REQUIREMENTS:

- DATABASE-MYSQL
- OS-FEDORA 20

#### THEORY:

##### JOINS:

A join is a method of linking data between one (self-join) or more tables based on values of the common column between the tables.

##### TYPES OF JOINS:

MySQL supports the following types of joins:

1. NATURAL JOIN
2. INNER JOIN
3. LEFT JOIN
4. RIGHT JOIN
5. CROSS JOIN

##### NATURAL JOIN:

Natural join does not use any comparison operator. It does not concatenate the way a Cartesian product does. We can perform a Natural Join only if there is at least one common attribute that exists between two relations. In addition, the attributes must have the same name and domain.

Natural join acts on those matching attributes where the values of attributes in both the relations are same.

SYNTAX-

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

INNER JOIN:

The INNER JOIN matches each row in one table with every row in other tables and allows you to query rows that contain columns from both tables.

The INNER JOIN is an optional clause of the SELECT statement. It appears immediately after the FROM clause. Here is the syntax of the INNER JOIN clause:

SYNTAX-

```
SELECT * FROM table1 INNER JOIN table2 ON join_condition1 INNER JOIN table3 ON  
join_condition2 ...;
```

LEFT JOIN:

The LEFT JOIN allows you to query data from two or more tables. Similar to the INNER JOIN clause, the LEFT JOIN is an optional clause of the SELECT statement, which appears immediately after the FROM clause.

SYNTAX-

```
SELECT * FROM table1 LEFT JOIN table2 ON join_condition;
```

RIGHT JOIN:

MySQL RIGHT JOIN is similar to LEFT JOIN, except that the treatment of the joined tables is reversed.

SYNTAX-

```
SELECT * FROM table1 RIGHT JOIN table2 ON join_condition;
```

CROSS JOIN:

The CROSS JOIN clause returns the Cartesian product of rows from the joined tables. The result set will include all rows from both tables, where each row is the combination of the row in the first table with the row in the second table. In general, if each table has  $n$  and  $m$  rows respectively, the result set will have  $n \times m$  rows.

## SYNTAX-

SELECT \* FROM table1 CROSS JOIN table2 ;

## SUB-QUERY:

A Subquery or Inner query or a Nested query is a query within another SQL query and embedded within the WHERE clause.

A subquery is used to return data that will be used in the main query as a condition to further restrict the data to be retrieved.

Subqueries can be used with the SELECT, INSERT, UPDATE, and DELETE statements along with the operators like =, <, >, >=, <=, IN, BETWEEN, etc.

## VIEWS:

Views are virtual tables that do not store any data of their own but display data stored in other tables. In other words, VIEWS are nothing but SQL Queries. A view can contain all or a few rows from a table. A MySQL view can show data from one table or many tables.

## SYNTAX-

CREATE VIEW view\_name AS SELECT STATEMENT;

## Types of views:

### 1. Simple View-

Simple view is view created on single table. When user wants data or some columns from same table then simple view is used. For an example if there is employee table which has Employee\_num, Employee\_name, salary columns and we just need to see Employee\_num, Employee\_name then user can create a simple view.

### 2. Complex View-

Complex view is view created on more than 1 tables. Complex view is created on using more than one tables. When user wants to retrieve data from more than 1 table then we have to use complex views. To create complex view there should be relation between 2 tables else cartesian product will come by joining 2 tables. There should be some joining conditions, some filters need to be considered while creating complex views.

## PURPOSE OF VIEWS:

- Security-

You can restrict users to access directly to a table and allow them to access a subset of data via views. For example, you can allow users to access customer name, phone, email via a view but restrict them to access the bank account and other sensitive information.

- Simplicity-

A relational database may have many tables with complex relationships e.g., one-to-one and one-to-many that make it difficult to navigate. However, you can simplify the complex queries with joins and conditions using a set of views.

- Consistency-

Sometimes, you need to write a complex formula or logic in every query. To make it consistent, you can hide the complex queries logic and calculations in views. Once views are defined, you can reference the logic from the views rather than rewriting it in separate queries.

#### COMMANDS OF VIEWS:

- CREATE VIEW-

```
CREATE VIEW view_name AS SELECT column1, column2, ... FROM table_name WHERE condition;
```

- INSERT IN VIEW-

```
INSERT INTO view_name ( field1, field2,...fieldN )VALUES(value1, value2,...valueN );
```

- UPDATE VIEW-

```
UPDATE view_name AS SELECT column1, column2, ... FROM table_name WHERE condition;
```

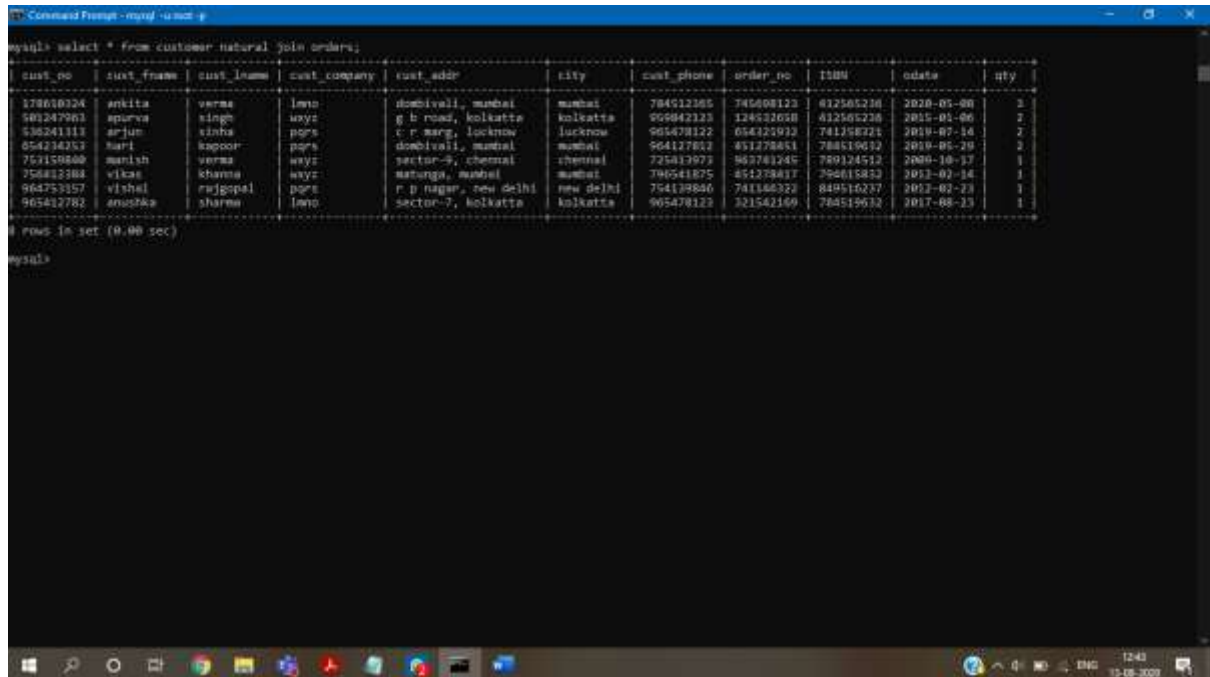
- DELETE FROM VIEW-

```
DELETE view_name AS SELECT column1, column2, ... FROM table_name WHERE condition;
```

- DROP VIEW-

DROP VIEW view\_name;

Find Customer details and order details using NATURAL JOIN.



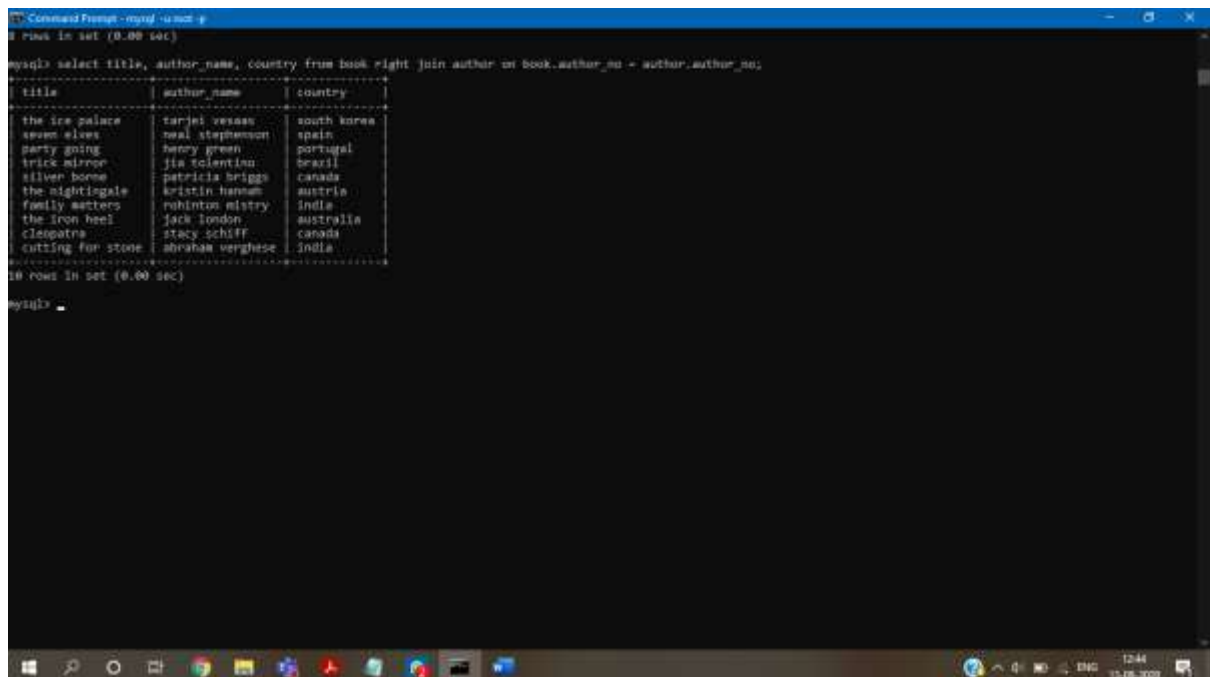
The screenshot shows a MySQL command prompt window with the following command and result:

```
mysql> select * from customer natural join orders;
```

cust_no	cust_fname	cust_lname	cust_company	cust_addr	city	cust_phone	order_no	TBNM	odate	qty
178610324	ankita	verma	lms	dombiwli, mumbai	mumbai	784512365	745608123	612565236	2020-05-08	2
501247961	apurva	singh	wxyz	g b road, kolkata	kolkata	905842323	124532658	612565236	2015-01-06	2
536241111	arjun	sinha	pqr	c r marg, lucknow	lucknow	965478122	054321932	741258221	2019-07-16	2
054234253	hari	kapoor	pqr	dombiwli, mumbai	mumbai	964127812	451278651	788519632	2019-05-29	2
753150840	manish	verma	wxyz	sector-9, chennai	chennai	725412972	962741245	789124512	2000-10-17	1
708823884	vikas	sharma	wxyz	satyaga, mumbai	mumbai	780541825	454278417	254819632	2012-02-14	1
964753357	vishal	rajshekar	pqr	r p nagar, new delhi	new delhi	754138846	741156322	849516237	2012-02-23	1
965412782	anushka	sharma	lms	sector-7, kolkata	kolkata	965478123	321542169	788519632	2017-08-23	1

8 rows in set (0.00 sec)

Find the book\_title, author\_name, country.



The screenshot shows a MySQL command prompt window with the following command and result:

```
mysql> select title, author_name, country from book right join author on book.author_no = author.author_no;
```

title	author_name	country
the ice palace	tarjet vesna	south korea
seven elves	neal stephenson	spain
party going	henry green	portugal
trick mirror	jia solentian	brasil
silver borne	patricia briggs	canada
the nightingale	kristin hannan	austria
family matters	rohinton mistry	india
the iron heel	jack london	australia
cleopatra	stacy schiff	canada
cutting for stone	abraham verghese	india

10 rows in set (0.00 sec)

Find the customer ID, name and order\_no of customers who have never placed an order.

```
mysql> select customer.cust_no, cust_fname, cust_lname, order_no from customer left join orders on customer.cust_no = orders.cust_no where order_no is null;
```

cust_no	cust_fname	cust_lname	order_no
754113884	puneet	singh	NULL
754177184	raahis	chopra	NULL

```
2 rows in set (0.00 sec)
```

```
mysql>
```

Find the Title, ISBN, order\_no of the books for which order is not placed.

```
mysql> select title, book.ISBN, order_no from orders right join book on book.ISBN = orders.ISBN where order_no is null;
```

title	ISBN	order_no
cutting for stone	049541237	NULL
cleopatra	063214785	NULL

```
2 rows in set (0.00 sec)
```

```
mysql>
```

Display cust\_fname, title,author\_no,publisher\_year where ISBN=1234.

```
Command Prompt - mysql -u root -p
Row matched: 1 Changed: 1 Warnings: 0

mysql> select cust_fname, title, author_no, pub_year from customer left join orders on customer.cust_no = orders.cust_no right join book on book.ISBN = orders.ISBN where
book.ISBN = 1234;
+-----+-----+-----+-----+
| cust_fname | title          | author_no | pub_year |
+-----+-----+-----+-----+
| vikas      | family matters | 791135045 | 2004      |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Display the total number of books and customer name.

```
Command Prompt - mysql -u root -p

mysql> select count(cust_fname),count(title) from customer left join orders on customer.cust_no = orders.cust_no right join book on book.ISBN = orders.ISBN;
+-----+-----+
| count(cust_fname) | count(title) |
+-----+-----+
| 8                  | 12            |
+-----+-----+
1 row in set (0.00 sec)

mysql>
```



List the cust\_id, order\_no and ISBN with books having title 'mysql'.

```
mysql> select order_no, cust_no, orders.ISBN from orders left join book on orders.ISBN = book.ISBN where title like 'mysql';
```

order_no	cust_no	ISBN
828008267	965474112	941287448

```
1 row in set (0.00 sec)
```

```
mysql>
```

Find the names of all the companies that ordered books in the year 2015.

```
mysql> select cust_company from customer left join orders on customer.cust_no = orders.cust_no where odate like '2015%';
```

cust_company
Wxyz

```
1 row in set, 1 warning (0.00 sec)
```

```
mysql>
```

Create view showing the author and book details.

```
mysql> create view author_book as
  -> select author_name, country, title, unit_price, publisher_no, pub_year from author
  -> right join book on book.author_no = author.author_no;
Query OK, 0 rows affected (0.28 sec)

mysql> select * from author_book;
+-----+-----+-----+-----+-----+-----+
| author_name | country | title | unit_price | publisher_no | pub_year |
+-----+-----+-----+-----+-----+-----+
| kristin hannah | austria | the nightingale | 580 | 159487263 | 2008 |
| henry green | portugal | party going | 450 | 054321887 | 1938 |
| jack london | australia | the iron heel | 295 | 748596123 | 2006 |
| neal stephenson | spain | seven elves | 578 | 142536789 | 2015 |
| jia tolentino | brazil | trick mirror | 479 | 468213579 | 2019 |
| rohinan elstey | india | family matters | 450 | 497851236 | 2004 |
| tarjel vestas | south korea | the ice palace | 568 | 741258147 | 1963 |
| abraham verghese | india | cutting for stone | 595 | 748159263 | 2009 |
| stacy schiff | canada | cleopatra | 195 | 963751482 | 2000 |
| patricia briggs | canada | silver burne | 720 | 482637150 | 2000 |
+-----+-----+-----+-----+-----+-----+
10 rows in set (0.01 sec)

mysql>
```

Perform Manipulation on simple view-Insert, update, delete, drop view.

```
mysql> create view personal_details as
--> select cust_fname, cust_lname, cust_addr, cust_phone
--> from customers;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from personal_details;
+-----+-----+-----+-----+
| cust_fname | cust_lname | cust_addr | cust_phone |
+-----+-----+-----+-----+
| ankita    | verma     | domblvali, mumbai | 784512101 |
| apurva    | singh     | g h road, kolkatta | 959842123 |
| arjun     | sinha     | c r marg, lucknow | 965478122 |
| hari      | Kapoor    | domblvali, mumbai | 964127812 |
| munish    | verma     | sector-9, Chennai | 725413971 |
| purnest    | singh     | kothrud, pune | 784512963 |
| rekha     | Chopra    | Katraj, pune | 784512263 |
| vikas     | Khanna    | Matunga, Mumbai | 796541875 |
| Vishal    | Rajgopal  | R P Nagar, New Delhi | 754139846 |
| Anushka    | Sharma    | sector-7, kolkatta | 965478123 |
+-----+-----+-----+-----+
10 rows in set (0.05 sec)

mysql> select cust_fname, cust_lname, cust_addr, cust_phone, city
--> at line 1
mysql> create or replace view personal_details as
--> select cust_fname, cust_lname, cust_addr, cust_phone, city
--> from customers;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from personal_details;
+-----+-----+-----+-----+-----+
| cust_fname | cust_lname | cust_addr | cust_phone | city |
+-----+-----+-----+-----+-----+
| ankita    | verma     | domblvali, mumbai | 784512101 | mumbai |
| apurva    | singh     | g h road, kolkatta | 959842123 | kolkatta |
| arjun     | sinha     | c r marg, lucknow | 965478122 | lucknow |
| hari      | Kapoor    | domblvali, mumbai | 964127812 | mumbai |
| munish    | verma     | sector-9, Chennai | 725413971 | Chennai |
| purnest    | singh     | kothrud, pune | 784512963 | pune |
| rekha     | Chopra    | Katraj, pune | 784512263 | pune |
| vikas     | Khanna    | Matunga, Mumbai | 796541875 | Mumbai |
| Vishal    | Rajgopal  | R P Nagar, New Delhi | 754139846 | New Delhi |
| Anushka    | Sharma    | sector-7, kolkatta | 965478123 | kolkatta |
+-----+-----+-----+-----+-----+
10 rows in set (0.01 sec)

mysql> create or replace view personal_details as
--> select cust_fname, cust_lname, cust_addr, cust_phone
--> from customers;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from personal_details;
+-----+-----+-----+-----+
| cust_fname | cust_lname | cust_addr | cust_phone |
+-----+-----+-----+-----+
| ankita    | verma     | domblvali, mumbai | 784512101 |
| apurva    | singh     | g h road, kolkatta | 959842123 |
| arjun     | sinha     | c r marg, lucknow | 965478122 |
| hari      | Kapoor    | domblvali, mumbai | 964127812 |
| munish    | verma     | sector-9, Chennai | 725413971 |
| purnest    | singh     | kothrud, pune | 784512963 |
| rekha     | Chopra    | Katraj, pune | 784512263 |
| vikas     | Khanna    | Matunga, Mumbai | 796541875 |
| Vishal    | Rajgopal  | R P Nagar, New Delhi | 754139846 |
| Anushka    | Sharma    | sector-7, kolkatta | 965478123 |
+-----+-----+-----+-----+
10 rows in set (0.01 sec)

mysql> drop view personal_details;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

#### CONCLUSION:-

We successfully understood types of joins , subquery and its types and complex views and implemented MYSQL Queries for suitable database application.