# High Impact Skills Development Program

# in Artificial Intelligence, Data Science, and Blockchain

# Lab 05: Set Operators & Views in SQL

# Instructor: Dr. Bilal

# Lab Engineer:

# SQL SET Operators & Views

# 

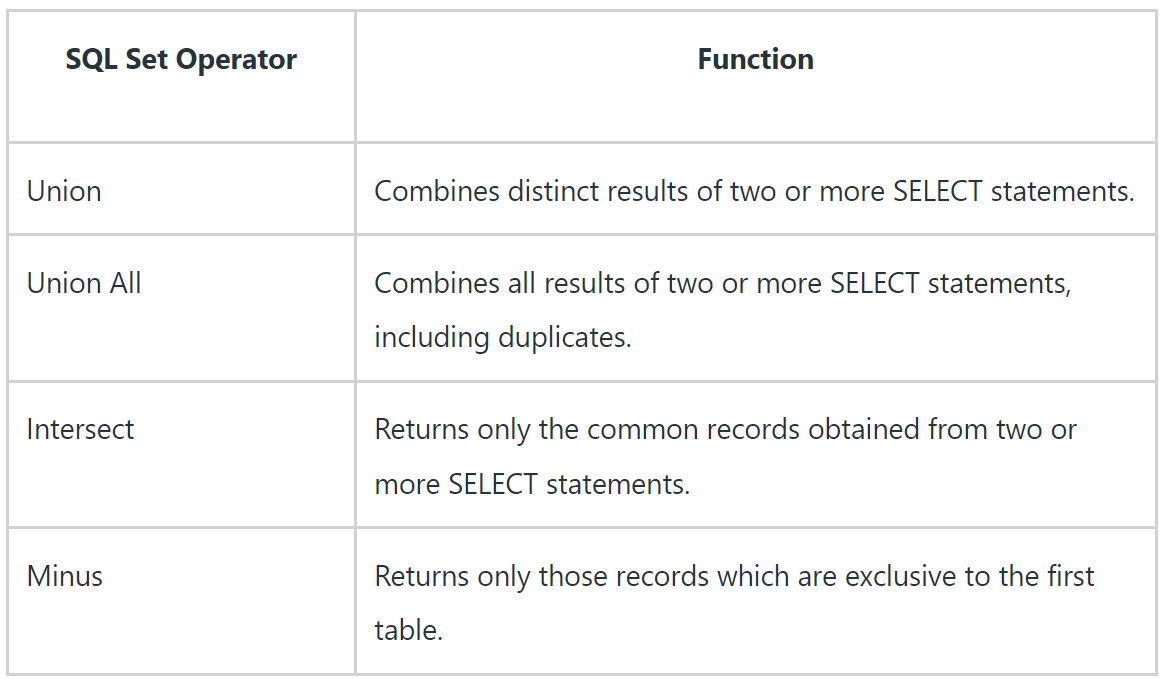
**Introduction**

**SET Operators:**

SQL set operators are used to combine the results obtained from two or more queries into a single result. The queries which contain two or more subqueries are known as compounded queries.

There are four major types of SQL operators, namely:

1. Union
2. Union all
3. Intersect
4. Minus/EXCEPT

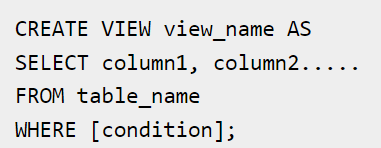


**Views:**

A view can contain rows from an existing table (all or selected). A view can be created from one or many tables which depends on the written SQL query to create a view. Unless indexed, a view does not exist in a database.

Views, which are a type of virtual tables allow users to do the following −

* Structure data in a way that users or classes of users find natural or intuitive.
* Restrict access to the data in such a way that a user can see and (sometimes) modify exactly what they need and no more.
* Summarize data from various tables which can be used to generate reports.



**Objectives**

After performing this lab students should be able to: 1. Design SQL queries to create views, manipulating and retrieving data from views 2. Perform analysis through set operators.

**Tools/Software Requirement**  
• MySQL Community Server 5.6

• MySQL Workbench 6.1

• Sakila Database

**Lab Task**

**Views:**

1. Create a table customers:

CREATE TABLE CUSTOMERS(

ID INT NOT NULL,

NAME VARCHAR (20) NOT NULL,

AGE INT NOT NULL,

ADDRESS CHAR (25) ,

SALARY DECIMAL (18, 2),

PRIMARY KEY (ID)

);

1. Now, insert values into this table using the INSERT statement as follows −

INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)

VALUES (1, 'Ramesh', 32, 'Ahmedabad', 2000.00 );

INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)

VALUES (2, 'Khilan', 25, 'Delhi', 1500.00 );

INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)

VALUES (3, 'kaushik', 23, 'Kota', 2000.00 );

INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)

VALUES (4, 'Chaitali', 25, 'Mumbai', 6500.00 );

INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)

VALUES (5, 'Hardik', 27, 'Bhopal', 8500.00 );

INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)

VALUES (6, 'Komal', 22, 'MP', 4500.00 );

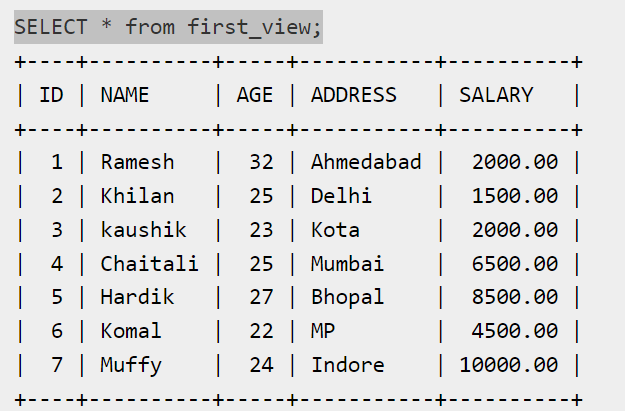
INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)

VALUES (7, 'Muffy', 24, 'Indore', 10000.00 );

1. Following query creates a view based on the above created table –

Create view first\_view AS SELECT \* FROM customers;

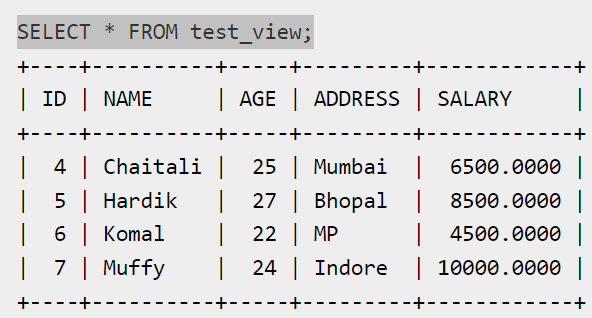
1. You can verify the contents of a view using the select query as shown below –



1. create a view using the where clause as shown below −

CREATE VIEW test\_view as SELECT \* FROM customers where SALARY>3000;

1. You can verify the contents of a view using the select query as shown below –



**Set Operators:**

1. Create Table Order using following SQL query-

CREATE TABLE ORDERS (

OID INT NOT NULL,

DATES DATETIME NOT NULL,

CUSTOMER\_ID INT NOT NULL,

AMOUNT INT NOT NULL,

PRIMARY KEY (OID)

);

1. Inserts values into this table using the INSERT statement –

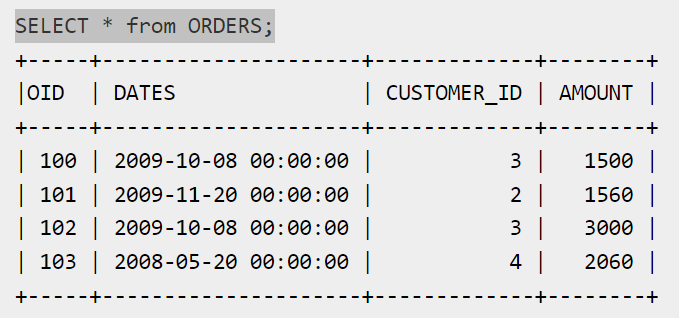
insert INTO ORDERS VALUES(102, '2009-10-08 00:00:00', 3, 3000);

insert INTO ORDERS VALUES(100, '2009-10-08 00:00:00', 3, 1500);

insert INTO ORDERS VALUES(101, '2009-11-20 00:00:00', 2, 1560);

insert INTO ORDERS VALUES(103, '2008-05-20 00:00:00', 4, 2060);

1. Verify the contents of the ORDERS table



1. Try to combine the SALARY and AMOUNT columns from CUSTOMERS and ORDERS respectively

SELECT SALARY

FROM CUSTOMERS

UNION

SELECT AMOUNT

FROM ORDERS

1. Find common the SALARY and AMOUNT columns from CUSTOMERS and ORDERS respectively

SELECT SALARY

FROM CUSTOMERS

INTERSECT

SELECT AMOUNT

FROM ORDERS

1. Find unique of the Salary not matched AMOUNT columns from CUSTOMERS and ORDERS respectively

SELECT SALARY

FROM CUSTOMERS

EXCEPT

SELECT AMOUNT

FROM ORDERS