# LAB 6 - VIEW

## I. SQL CREATE VIEW Statement

In SQL, a VIEW is a virtual table based on the result-set of an SQL statement.

A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.

You can add SQL functions, WHERE, and JOIN statements to a view and present the data as if the data were coming from one single table.

### Syntax

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

**Note:** A view always shows up-to-date data! The database engine recreates the data, using the view's SQL statement, every time a user queries a view.

## II. The SQL UPDATE VIEW command can be used to modify the data of a view.

All views are not updatable. So, UPDATE command is not applicable to all views. An updatable view is one which allows performing a UPDATE command on itself without affecting any other table.

### When can a view be updated?

1. The view is defined based on one and only one table.

2. The view must include the PRIMARY KEY of the table based upon which the view has been created.

3. The view should not have any field made out of aggregate functions.

4. The view must not have any DISTINCT clause in its definition.

5. The view must not have any GROUP BY or HAVING clause in its definition.

6. The view must not have any SUBQUERIES in its definitions.

7. If the view you want to update is based upon another view, the later should be updatable.

8. Any of the selected output fields (of the view) must not use constants, strings or value expressions.

### Syntax:

```
UPDATE < view_name > SET <column1>=<value1>, <column2>=<value2>,.....
WHERE <condition>;
```

## III. SQL Dropping a View

You can delete a view with the DROP VIEW command.

### Syntax

```
DROP VIEW view_name;
```

# Exercises

Given a database below.

`Salesman (Salesman_ID: numberic(5); Name: varchar(30); City: varchar(15); Commission: decimal(5,2))`

`Orders (ord_No: numeric(5); purch_AMT: decimal(8,2); ord_date: date; customer_ID: numeric(5); salesman_ID: numeric(5))`

`Customer (customer_ID: numeric(5); cust_name: varchar(30); city: varchar(15); grade: numeric(3); salesman_ID: numeric(5))`

NOTE: The data of each table will be imported from attached files.

1. Write a query to create a view for those salesmen belongs to the city New York.

Sample answer:

```
CREATE VIEW newyorkstaff
AS SELECT *
FROM salesman
WHERE city = 'New York';
```

2. Write a query to create a view for all salesmen with columns salesman_id, name, and city.
3. Write a query to find the salesmen of the city New York who achieved the commission more than 13%.
4. Write a query to create a view to getting a count of how many customers we have at each level of a grade.
5. Write a query to create a view to keeping track the number of customers ordering, number of salesmen attached, average amount of orders and the total amount of orders in a day.

6. Write a query to create a view that shows for each order the salesman and customer by name.
7. Write a query to create a view that finds the salesman who has the customer with the highest order of a day.
8. Write a query to create a view that finds the salesman who has the customer with the highest order at least 3 times on a day.
9. Write a query to create a view that shows all of the customers who have the highest grade.
10. Write a query to create a view that shows the number of the salesman in each city.
11. Write a query to create a view that shows the average and total orders for each salesman after his or her name. (Assume all names are unique)
12. Write a query to create a view that shows each salesman with more than one customers.
13. Write a query to create a view that shows all matches of customers with salesman such that at least one customer in the city of customer served by a salesman in the city of the salesman.
14. Write a query to create a view that finds the salesmen who issued orders on either August 17th, 2017 or October 10th, 2017.